CS 5350/6350: Machine Learining Spring 2018

Homework 3 Solutions

Yulong Liang (u1143816)

April 4, 2018

1 Simple Conjunction Learner Analysis

1. [50 points]

   (a) $n = 100,\ 1 - \delta = 0.9,\ \epsilon = 0.1$

   $$m > \frac{n}{\epsilon}\Big(\ln(n) + \ln(\frac{1}{\delta})\Big) = \frac{100}{0.1}\Big(\ln(100) + \ln(\frac{1}{0.1})\Big) \approx 6907.8$$

   We will need at least $6,908$ examples.

   (b) $n = 100,\ 1 - \delta = 0.99,\ \epsilon = 0.1$

   $$m > \frac{n}{\epsilon}\Big(\ln(n) + \ln(\frac{1}{\delta})\Big) = \frac{100}{0.1}\Big(\ln(100) + \ln(\frac{1}{0.01})\Big) \approx 9210.3$$

   We will need at least $9,211$ examples.

   (c) $n = 100,\ 1 - \delta = 0.9,\ \epsilon = 0.01$

   $$m > \frac{n}{\epsilon}\Big(\ln(n) + \ln(\frac{1}{\delta})\Big) = \frac{100}{0.01}\Big(\ln(100) + \ln(\frac{1}{0.1})\Big) \approx 69077.6$$

   We will need at least $69,078$ examples.

   (d) $n = 100,\ 1 - \delta = 0.99,\ \epsilon = 0.01$

   $$m > \frac{n}{\epsilon}\Big(\ln(n) + \ln(\frac{1}{\delta})\Big) = \frac{100}{0.01}\Big(\ln(100) + \ln(\frac{1}{0.01})\Big) \approx 92103.4$$

   We will need at least $92,104$ examples.

   (e) $n = 1000,\ 1 - \delta = 0.99,\ \epsilon = 0.01$

   $$m > \frac{n}{\epsilon}\Big(\ln(n) + \ln(\frac{1}{\delta})\Big) = \frac{1000}{0.01}\Big(\ln(1000) + \ln(\frac{1}{0.01})\Big) \approx 1151292.5$$

   We will need at least $1,151,293$ examples.

2. [10 points] Show that for $0 \leq x \leq 1$,

$$1 - x \leq e^{-x}.$$

Proof Let $f(x) = e^{-x} - (1-x) = e^{-x} + x - 1$.

$$\text{The derivative } f'(x) = -e^{-x} + 1$$

$$\text{The second derivative } f''(x) = -(-e^{-x}) = e^{-x}$$
$$\therefore f''(x) = e^{-x} > 0, x \in [0,1]$$
$$\therefore f'(x) \text{ increases within } [0,1]$$
$$f'(0) = -e^{-0} + 1 = -1 + 1 = 0$$
$$\therefore f'(x) \geq 0, x \in [0,1]$$
$$\therefore f(x) \text{ increases within } [0,1]$$
$$f(0) = e^{-0} + 0 - 1 = 1 - 1 = 0$$
$$\therefore f(x) \geq 0, x \in [0,1]$$
$$\therefore e^{-x} \geq 1 - x$$

2 Occam's Razor and PAC guarantee for consistent learners

1. [20 points]

   (a) I will prefer L_2. According to Occam's Razor principle, we should refer simpler explanations over more complex ones. Since both algorithms can find a hypothesis consist with the training data, I will prefer L_2 because it has a smaller hypothesis space.

   (b) According to the about inequality, the number of training examples needed **m** is proportional to the logarithm of size of hypothesis space $|H|$, logarithm of reciprocal of probability of failure δ and reciprocal of error level ϵ. For a fixed probability level $1-\delta$ and a fixed error level ϵ, the larger hypothesis space is, the more training examples are required.

3 PAC Learnable Results

1. [10 points] General disjunctions out of n binary variables.

   - size of hypothesis space:
   $$|H| = 3^n$$
   - number of training examples needed:
   $$m > \frac{1}{\epsilon}\Big(\ln(|H|) + ln(\frac{1}{\delta})\Big) = \frac{1}{\epsilon}\Big(n\ln(3) + ln(\frac{1}{\delta})\Big)$$

The number of training examples needed m is polynomial to $\frac{1}{\epsilon}$, $\frac{1}{\delta}$, n, and $|H|$. Thus general disjunctions out of n binary variables is **PAC learnable**.

2. [10 points] m-of-n rules (Note that m is a fixed constant).

   - size of hypothesis space:

   $$|H| = m\binom{n}{m} \approx mn^m$$

   - number of training examples needed:

   $$m > \frac{1}{\epsilon}\Big(\ln(|H|) + ln(\frac{1}{\delta})\Big) = \frac{1}{\epsilon}\Big(\ln(m) + m\ln(n) + ln(\frac{1}{\delta})\Big)$$

   The number of training examples needed m is polynomial to $\frac{1}{\epsilon}$, $\frac{1}{\delta}$, n, and $|H|$. Thus m-of-n rules is **PAC learnable**.

3. [10 points] Simple conjunctions out of n binary variables.

   - size of hypothesis space:

   $$|H| = 2^n$$

   - number of training examples needed:

   $$m > \frac{1}{\epsilon}\Big(\ln(|H|) + ln(\frac{1}{\delta})\Big) = \frac{1}{\epsilon}\Big(n\ln(2) + ln(\frac{1}{\delta})\Big)$$

   The number of training examples needed m is polynomial to $\frac{1}{\epsilon}$, $\frac{1}{\delta}$, n, and $|H|$. Thus simple conjunctions out of n binary variables is **PAC learnable**.

4. [10 points] k-CNF out of n binary variables.

   - number of conjuncts:

   $$O((2n)^k)$$

   - size of hypothesis space:

   $$|H| = O(2^{(2n)^k})$$
   $$ln(|H|) = O((2n)^k \ln 2)$$

   - number of training examples needed:

   $$m > \frac{1}{\epsilon}\Big(\ln(|H|) + ln(\frac{1}{\delta})\Big) = O\Big(\frac{1}{\epsilon}\Big((2n)^k\ln(2) + ln(\frac{1}{\delta})\Big)\Big)$$

   The number of training examples needed m is polynomial to $\frac{1}{\epsilon}$, $\frac{1}{\delta}$, n, and $|H|$. Thus k-CNF out of n binary variables is **PAC learnable**.

5. [10 points] General boolean functions of n binary variables.

- size of hypothesis space:
$$|H| = 2^{2^n}$$
- number of training examples needed:
$$m > \frac{1}{\epsilon}\Big(\ln(|H|) + ln(\frac{1}{\delta})\Big) = \frac{1}{\epsilon}\Big(2^n \ln(2) + ln(\frac{1}{\delta})\Big)$$

The number of training examples needed m is exponential to n. Thus general boolean functions of n binary variables is **NOT PAC learnable**.